A quick tour through the available diagnostic and monitoring tools for White Rabbit Networks

Adam Wujek
Independent White Rabbit Consultant
dev_public at wujek dot eu
DISCLAIMER:
This presentation is an overview and shall be treated as a starting point rather than detailed manual. For details please refer to:
- Switch user manual
- Switch developer manual
- White Rabbit Switch: Failures and Diagnostics
- WRPC user manual
- Other manuals
White Rabbit Switch (WRS)
WRS: wr_mon

- The best tool to start with
- Gives many important information about Switch status and configuration:
  - Build version
    (of the tool! Not of the entire firmware.
    For the deployed firmware should not matter)
  - Link status
  - PTP/WR configuration
  - MAC of the peer (*)
  - PLL locking status
  - Timing mode
  - WR time
  - Servo status (if in slave mode)

(*) - not in the v6.0 release
WRS: wrs_shmem_dump

- Dump run-time internal data of:
  - wrsw_hal (HAL)
  - ppsi
  - wrsw_rtud (RTU)
  - soft PLL stats
- wrsw_hal, ppsi and wrsw_rtud store internal data in shared memory in /dev/shm/wrs-shmem-*
- In format <name>: <value>
- Useful when a requested information is not displayed by any other tool
WRS: rtu_stat

- Print details about:
  - Switching tables
  - VLANs
  - Port mirroring details
- Can also set the above

```
wrch1#rtu_stat
RTU Filtering Database Dump: 22 rules

<table>
<thead>
<tr>
<th>MAC</th>
<th>Dist.ports</th>
<th>FID</th>
<th>Type</th>
<th>Age [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>01:68:5e:80:00:6b</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>66:77</td>
</tr>
<tr>
<td>01:68:5e:80:01:01</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>61:03</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>ALL CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
<tr>
<td>01:68:5e:80:00:00</td>
<td>CPU</td>
<td>0</td>
<td>STATIC</td>
<td>00:00</td>
</tr>
</tbody>
</table>
```

RTU VLAN Table Dump:

<table>
<thead>
<tr>
<th>VID</th>
<th>FID</th>
<th>MASK</th>
<th>DROP</th>
<th>PRIOR</th>
<th>PRIOR OVERRIDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0x0007fff</td>
<td>NO</td>
<td>--</td>
<td>NO</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>0x0007ffe</td>
<td>NO</td>
<td>--</td>
<td>NO</td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>0x00040001</td>
<td>NO</td>
<td>--</td>
<td>NO</td>
</tr>
<tr>
<td>102</td>
<td>102</td>
<td>0x0007fff</td>
<td>NO</td>
<td>--</td>
<td>NO</td>
</tr>
</tbody>
</table>

4 active VIDs defined

RTU Port Mirroring Config Dump:

Status: Disabled
**WRS: wrs_vlans**

- `wrs_vlans --plist`  
  - Print ports’ VLAN configuration
- `wrs_vlans --list`  
  - Print VLAN configuration
- Useful in configuration and verification of VLANs

```
wrch2# wrs_vlans --plist
# HP mask: 0x00
#
# QMODE FIX_PRIO PRIO PVID MAC UNTAG
# wri1 1 TRUNK 0 0 0 000000000000 0
wri2 1 TRUNK 0 0 0 000000000000 0
wri3 1 TRUNK 0 0 0 000000000000 0
wri4 1 TRUNK 0 0 0 000000000000 0
wri5 1 TRUNK 0 0 0 000000000000 0
wri6 1 TRUNK 0 0 0 000000000000 0
wri7 1 TRUNK 0 0 0 000000000000 0
wri8 1 TRUNK 0 0 0 000000000000 0
wri9 1 TRUNK 0 0 0 000000000000 0
wri10 1 TRUNK 0 0 0 000000000000 0
wri11 1 TRUNK 0 0 0 000000000000 0
wri12 1 TRUNK 0 0 0 000000000000 0
wri13 1 TRUNK 0 0 0 000000000000 0
wri14 1 TRUNK 0 0 0 000000000000 0
wri15 1 TRUNK 0 0 0 000000000000 0
wri16 1 TRUNK 0 0 0 000000000000 0
wri17 1 TRUNK 0 0 0 000000000000 0
wri18 0 ACCESS 0 0 4093 000000000000 1
```
WRS: wrs_version

- Print details about switch’s hardware version
- Version of this tool is used as the version of a firmware (e.g. by SNMP, LLDP, webinterface)
- Provides the best answer for question “Which version do you use?”

```
#wrs_version -t
software-version: WP3a-wrpc_fixes-16-g1ed690ff-dirty
built-by: Adam Wujek
build-date: Sep 7 2021 14:46:54
backplane-version: 3.30
fpga-type: LX240T
manufacturer: [redacted]
serial-number: [redacted]
scb-version: 3.4
gateware-version: 6.0
gateware-build: 06/04/20.01
wr_switch_hdl-commit: 4e89257
general-cores-commit: dcc7cc3
wr-cores-commit: 8299d65
```
## WRS: wrs_pstats

- Print frame counters for wr ports
- Can print different sets of counters

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wr1</td>
<td>0</td>
<td>...</td>
<td>2285115</td>
<td>8490614</td>
<td>0</td>
<td>8490574</td>
<td>8490574</td>
<td>5900726</td>
<td>0</td>
<td>5900726</td>
<td>2590848</td>
<td>8490574</td>
<td>8490574</td>
<td>1104729</td>
</tr>
<tr>
<td>wr2</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr3</td>
<td>0</td>
<td>...</td>
<td>1930535</td>
<td>1378593</td>
<td>0</td>
<td>1378426</td>
<td>1378426</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr4</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr5</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr6</td>
<td>0</td>
<td>...</td>
<td>1931633</td>
<td>1379560</td>
<td>0</td>
<td>1379484</td>
<td>1379484</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr7</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr8</td>
<td>0</td>
<td>...</td>
<td>2483173</td>
<td>1103637</td>
<td>0</td>
<td>1103637</td>
<td>1103637</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr9</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr10</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr11</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr12</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr13</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr14</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr15</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr16</td>
<td>0</td>
<td>...</td>
<td>1380040</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr17</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wr18</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
WRS: wrs_sfp_dump

- Detailed information about used SFPs:
  - Vendor Name
  - Product Name
  - Serial

- Include SFP's monitoring data (DOM; if enabled in dot-config)
  - Temperature
  - Voltage
  - Bias Current
  - TX/RX power

- Can read data from HAL or directly from SFP
  - Reading directly via I2C, when HAL is running can corrupt SFP's EEPROM! Disable monit and HAL first!

- Can read and write SFP's EEPROM

```
wrch2#wrs_sfp_dump -p 3 -d -L
Reading SFP eeprom from HAL
======== port 3 ========
Identifier: 03
Extended Identifier: 04
Connector: 07
Transceiver: 00400000070403
Encoding: 01
Nominal Bit Rate: 1300 Megabits/s
Length (9m): 20km
Length (9m): 20000m
Length (50m): 0m
Length (62.5m): 0m
Length (copper): 0m
Vendor Name: FS
Company ID: 000000
Vendor Part Number: GE-LC-1490
Vendor Revision:
TX Wavelength: 1490
Options: IA00
Bitrate (MAX): 00
Bitrate (MIN): 00
Vendor Serial: C1904080513
Date Code: 190412
Temperature: 46.520 C
Voltage: 3.208 V
Bias Current: 14.624 mA
TX power: 0.233 mW
RX power: 0.331 mW
```
WRS: /tmp

- Various files in /tmp:
  - custom_boot_script_status
  - dot-config_status
  - hwinfo_read_status
  - leapseconds_check_status
  - vlans_set_status
  - wrs_auxclk_set_status
  - wrs_throttling_set_status
  - load_fpga_status
  - load_lm32_status
  - dot-config_source
  - leapseconds_download_source
  - start_cnt_httpd
  - start_cnt_lldpd
  - start_cnt_rvlan
  - start_cnt_snmpd
  - start_cnt_sshd
  - start_cnt_syslogd
  - start_cnt_wrs_watchdog
  - syslog
WRS: ptpdump

- Frame/packet sniffer
- Dumps PTP frames on a given interface
- Useful in verification of the link status and configuration (PTP)
WRS: tcpdump & wireshark

- tcpdump - network sniffer
- wireshark - network sniffer with GUI
  - can dissect WR specific fields in PTP frames/packets since version 2.9
- Version available in Ubuntu 20.04 LTS
- tcpdump can be combined with wireshark to dissect frames in live over ssh:

  ```
  ssh root@wrs "tcpdump -i wri1 --immediate-mode --packet-buffered -w -" | wireshark -i -
  ```
WRS: wrs_dump.sh

• Dump the current state of WRS
• Run over ssh from the host
• Not in the release, but can be used with old releases
• Script gets the following:
  - output of wrs_version
  - output of w command (logged users and uptime)
  - process list
  - output of wrs_dump_shmem
  - output of wrs_pstats
  - output of wr_mon
  - output of df
  - output of free
  - output of /proc/meminfo
  - output of ifconfig
  - tcpdump of up ports (on specified ports depending on the parameter)
  - output of PPSI's verbose messages (if selected by the parameter)
  - output of dmesg
  - output of wrs_vlans --list
  - output of wrs_vlans --plist
  - output of rtu_stat
  - output of wrs_sfp_dump -L -d -x
  - dot-config
  - shmem files
  - content of /tmp

• Gets the output of some commands twice

$ ./wrs_dump.sh root@wrs
Open ssh connection...
Provide password
ssh connection established
Store data in the directory wrs_dump-wrs-2021-10-05_03-4... Done
Get version... Done
Get w (logged users and uptime)... Done
Get process list... Done
Get output of wrs_dump_shmem... Done
Get output of wrs_pstats... Done
Get output of wr_mon... Done
Get output of df... Done
Get output of free... Done
Get output of /proc/meminfo... Done
Get output of ifconfig... Done
Get output of dmesg... Done
Get output of wrs_vlans --list... Done
Get output of wrs_vlans --plist... Done
Get output of rtu_stat... Done
Get output of wrs_sfp_dump -L -d -x... Done
Copy dot-config... Done
Copy shmem... Done
Copy /tmp... Done
Get again process list... Done
Get again output of wrs_dump_shmem... Done
Get again output of wrs_pstats... Done
Get again output of wr_mon... Done
Get again output of df... Done
Get again output of free... Done
Get again output of /proc/meminfo... Done
Get again output of ifconfig... Done
Closing ssh connection... Done
WRS: SNMP

- WR-SWITCH-MIB
  - status OIDs
  - expert OIDs

More in "White Rabbit Switch: Failures and Diagnostics"

- standard MIBs (not in v6.0 release):
  - MIB-IP
  - Q-BRIDGE-MIB
  - BRIDGE-MIB
WRS: LLDP

- Displays details of a device on the other side of a link

![Source: https://lldpd.github.io/](https://lldpd.github.io/)

```
wrch2#lldpcli show

LLDP neighbors:
Interface: wrch, V1a: LLDP, RID: 1, Time: 0 days, 02:18:30
ChassisID: mac [redacted] 92
SysName: 3com 9200 cisco switch
SysDescr: 3com 9200 cisco switch
PortID: 1.1.110
PortDescr: GigabitEthernet1/0/4 Interface
TTL: 120

Interface: wr13, V1a: LLDP, RID: 2, Time: 0 days, 02:18:30
ChassisID: mac [redacted] 92:71
SysName: wrch1
MgmtIP: 192.168.1.110
MgmtFacet: 2
Capability: Bridge, off
Capability: PortNet, off
Capability: Station, on
PortID: [redacted]
PortDescr: wr11
TTL: 20
```
WRS: radius

- Based on information from RADIUS server, WRS can limit access to WR network
- Uses VLANs to limit access
- Not in the release v6.0
WRS: syslog

- Gathers logs from all daemons
- Can store locally
- Can send to external server
White Rabbit PTP Core (WRPC)
WRPC: communication

WR network

WR port

PCIe

WRC#

FPGA

Connector
WRPC: Notes

NOTE:

- Most of these features can be enabled/disabled in dot-config
- Some are not present in the current release (4.2), but are planned for the coming release (v5.0)
WRPC: gui

- The best tool to start with
- Equivalent of wr_mon for WRPC
- Redesigned for WRPC v5
- Gives many important information about WRPC status and configuration:
  - Build version
  - Link status
  - PTP/WR configuration
  - MAC of the peer
  - PLL locking status
  - Timing mode
  - WR time
  - Servo status (if in slave mode)
WRPC: stat

- Prints to the console basic information about WRPC’s state
  - At the state change or every second
WRPC: wr_diags

• Uses wishbone registers to store basic parameters
• Run on a host

```
root@test1:~/.spec# ./wrpc-diags -f /sys/devices/pci0000:00/0000:00:1c.0/0000:20:00.0/resource0 -o 0x40800
Wishbone register version: in FPGA = 0x1 | in SW = 0x1

test1:wrcdiag[00] > diags
servo status:  Track phase
Port status:   Link up, PLL locked,
PTP state:    PPS slave
Aux state:    
TX frame count: 14682
RX frame count: 20397
TAI time:     Wed Oct  6 01:55:18 2021
Round trip time: 959130 ps
Master slave delay: 251904 ps
Total Link asymmetry: 34 ps
Clock offset: 2 ps
Phase setpoint: 15607 ps
Update counter: 2995
temp:         48.8125 C
```
WRPC: sfp

- Limited equivalent of wrs_sfp_dump
- Prints basic information about SFP

```
wrc# sfp info
Nominal Bit Rate: 1300 Mbits/s
Vendor Name: FS
Vendor PN: GE-LC-1310
Vendor serial: C2003129746
TX Wavelength: 1310
Temperature: 36.97 C
Voltage: 3.3488 V
Bias Current: 22.88 mA
TX power: 0.2265 mW
RX power: 0.2856 mW
```

- Manages local SFP database in non volatile memory

```
wrc# sfp show
1: PN:test PN1    dTx: 11112 dRx: 11113 alpha: 47734266527744
2: PN:test PN2    dTx: 99991 dRx: 99992 alpha: 429466664828928
3: PN:test PN3    dTx: 33334 dRx: 33335 alpha: 143177029779456
```
WRPC: ps

- Prints:
  - Task list
  - Total time spent on task
  - Maximum time spent on task
- Possible to reset counters
- If any task is executed longer than 250-500ms might cause problems with synchronization. This tool can help detect this
WRPC: config

- Prints configuration (dot-config) used at compile time
- Informs whether particular feature is enabled
- As disadvantage, it takes more than 4KB of memory.

```bash
wrc# config
   Current WRPC-SW configuration:
   CONFIG_ARCH_LM32=y
   # CONFIG_ARCH_RISCV is not set
   CONFIG_TARGET_GENERIC_PHY_8BIT=y
   # CONFIG_TARGET_GENERIC_PHY_16BIT is not set
   # CONFIG_TARGET_WR_SWITCH is not set
   # CONFIG_TARGET_AFCZ is not set
   # CONFIG_TARGET_ERTM14 is not set
   # CONFIG_TARGET_SIS8300KU is not set
   # CONFIG_TARGET_PXIE_FMC is not set
   CONFIG_WR_NODE=y
   CONFIG_STACKSIZE=2048
   CONFIG_PRINT_BUFSIZE=256
   CONFIG_RAMSIZE=196608
   CONFIG_TEMP_POLL_INTERVAL=15
   CONFIG_TEMP_HIGH_THRESHOLD=70
   CONFIG_TEMP_HIGH_RAPPEL=60
   # CONFIG_PLL_VERBOSE is not set
   # CONFIG_WRC_VERBOSE is not set
```
WRPC: init

- Display commands to be run at startup
- Commands can be both:
  - Defined at compile time (in dot-config)
  - Stored in non-volatile memory (flash)
- Can help solving strange/unexpected behavior at start-up. E.g. after an update link not working due to (not) configured VLAN. Or if some commands are stored in the flash.

```
wrcl# init show
-- built-in script --
vlan off
help
-- user-defined script --
(empty)
```
WRPC: wrpc_dump

- Equivalent of *wrs_shmem_dump*
- Run on host:
  - Directly via PCIe (does not work at the moment)
  - On memory image gathered with *mapper*
- In format `<name>`: `<value>`
- Useful when a requested information is not displayed by any other tool
- The dump includes:
  - IP/MAC details
  - Task list (*ps* command)
  - SFP info (including monitoring)
  - SPLL info
  - Dot-config
  - PPSI’s internal data
  - Temperature sensors data
WRPC: verbose

- Set verbose level of PPSI
- Very useful for PPSI debugging
-Verbose levels:

```
verbose 0000000
\|\|\|\|\|\|\|\|\|\|\|\|\|
   \|config
   \|\|ext
   \|\|\|bmc
   \|\|\|\|\|\|\|\|\|\|\|\|\|\|\|\|\|\|\|\|
     \|servo
     \|\|\|frames
     \|\|\|time
     \|fsm
```
WRPC: netconsole

- Mirrors the serial console via UDP packets over WR network
- Implemented, but not released yet
WRPC: LLDP

- In general displays what device is on the other side of a link
- In WRPC’s case, only reports its presence to a peer
WRPC: SNMP

- Limited comparing to WRS
- WR-WRPC-MIB
  - Version
  - Time
  - Temperature
  - SPLL status
  - PPSI status
  - SFP management (add, remove, details)
  - Init script management (add, erase)
  - Netconsole configuration
  - Remote shell command execution
WRPC: remote shell command execution via SNMP

- Send a shell command via SNMP over WR network
WRPC: syslog

- Send to external server
- Send important event (like sync lost)
- Can send all messages from console
Thank you!
Questions?